Science World Journal Vol 11 (No 1) 2016
www.scienceworldjournal.org
ISSN 1597-6343

NEDAC: A WORM COUNTERMEASURE MECHANISM

*[1] Muhammad Aminu Ahmad , [1]Abubakar S. Magaji and [1]Sani Dari

[1]Department of Mathematical Sciences, Faculty of Science,
Kaduna State University, Kaduna, Nigeria.

*Corresponding Author's Email: muhdaminu@kasu.edu.ng

ABSTRACT
This article presents an Internet worm countermeasure mechanism that uses DNS activities as a behavioural technique to detect worm propagation. The mechanism also uses a data-link containment solution to block traffic from an infected host. The concept has been demonstrated using a developed prototype and tested in a virtualised network environment. An empirical analysis of network worm propagation has been conducted to test the capabilities of the developed countermeasure mechanism. The results show that the developed mechanism is sensitive in containing Internet worms.

Keywords: Worm Detection, Malware, cyber defense

INTRODUCTION

The Internet has provided a medium for communication and sharing of information amongst people, businesses, governments and organisations. As a result, the Internet services must be kept continuous and secured from network and malware attacks. Malicious software (malware) is a generic term for any software that enters a computer system without the authorisation of the user to perform unwanted actions (Niemelä & Palomäki, 2013). Malware can be classified under a number of headings, including viruses, worms, trojans, spyware, adware, rootkits, drive-by downloads and other malicious and unwanted software. Self-propagating malware (termed a worm) is a particular class of malware that is highly virulent due to its self-spreading features. Fast scanning worms are particularly dangerous class of worms that self-propagate very rapid without the need for human interaction, particularly zero-day fast scanning worm that use a vulnerability that has not been patched or widely acknowledged at the point of an outbreak (Tidy et al., 2014). Internet worm outbreaks (e.g. Slammer, Code Red and Witty (Joukov & Chiueh, 2003)) have been experienced on the Internet, which caused disruption of services and significant financial losses to government, transportation and other institutions. Outbreaks of effective fast Internet worms can cause significant damage that involve financial losses ranging from millions to billions of US Dollars: $US2.6Bn for Code Red, $US1.2Bn for Slammer and circa $US11M for Witty (Fosnock, 2005).

A vulnerability can be exploited by a worm if it is network reachable, provides remote code execution, provides network access, and does not require human interaction once exploited (Tidy et al., 2014). Individual vulnerabilities can be researched through a number of online sources that provide details of identified vulnerabilities such as the Common Vulnerabilities and Exposures (CVE) (CVE, 2014) and National Vulnerability Database (NVD) (NVD, 2014) systems. These sources published vulnerabilities including the Microsoft RDP vulnerability (CVE-2012-0002) of 2012, and the ShellShock (CVE-2014-6271) and Drupal (CVE-2014-3704) vulnerabilities of 2014. Thus, the present threat of worm event remains clear.

A range of behavioural detection and suppression mechanisms has been reported in previously published security research work. However, there are limitations and shortcomings in the reported mechanisms. These involve ineffectiveness in detecting worms (Jyothsna et al., 2011), resource consumption, delay in deployment and detection (Garcia-Teodoro et al., 2009), management overhead and computational complexity, and in most cases the techniques only slow worm infections (Li et al., 2008). The previously reported research work can be categorized into signature-based and anomaly-based detection systems. The signature-based detection system maintains a database of signatures for previously known attacks and raises an alarm if a datagram in the network matches a signature in the database. Anomaly-based detection systems examine network traffic in order to build a profile of the normal behaviour and then raise an alarm for events that deviate from the normal profile. In contrast to signature-based systems, anomaly-based systems can detect new attacks and therefore capable of detecting zero-day worms. The focus of this article is to present an anomaly-based detection scheme that uses datagram-header information to identify the presence of a worm.

The remainder of the article is presented as follows. Section 2 presents related work on worm detection and containment systems. Section 3 presents the description of the developed countermeasure mechanism. Section 4 presents the experimental evaluation of the reported mechanism using a developed prototype. Section 5 discusses the results of the evaluation experiments conducted and Section 6 concludes the paper and discusses possible future work.

Related Work

A number of detection techniques were reported that identify the presence of a worm using datagram header information (Smith et al., 2009). Among these approaches are those that monitor source and destination IP addresses of datagrams, such as the work reported by Williamson (2002). Williamson (2002) proposed a detection and suppression technique that uses the source and destination IP addresses of a host making a request to detect an attack. The technique delays request from a host if it is new, otherwise it will be processed as normal. However, many fast scanning Internet worms (TCP-based) initiate connection

requests to randomly-generated IP addresses, which results in a number of failed connections. As a result, some approaches used the status of connection requests to detect worm behaviour such as the work of Jung et al. (2004), Weaver et al. (2004) and Rasheed et al. (2009). This technique uses the count of successful and failed connection attempts to determine the presence of worm scanning. Additionally, Gu et al. (2004) used a technique that correlates source and destination IP addresses and source and destination ports to detect fast scanning worms. The technique uses an algorithm termed Destination Source Correlation (DSC) that keeps track of SYN datagrams and UDP traffic of the source and destination. Thus, if a host received a datagram on port i, and then starts sending datagrams destined for port i, it becomes a suspect. Then if the immediate outgoing scan rate for the suspect host deviates from a normal profile, the host is considered to be infected.

Another detection approach is the use of DNS activities of hosts to detect worm propagation. Whyte et al. (2005) and Shahzad & Woodhead (2014a) used DNS-based rate limiting to suppress fast scanning Internet worms in an enterprise network. The observation was scanning worms often use numeric IP addresses instead of the qualified domain name of a system, which eliminates the need for a DNS query. In contrast, the vast majority of legitimate publicly available services are accessed through the use of DNS protocol; the network service that maps numeric IP addresses to corresponding alphanumeric names. Therefore the main idea behind this technique is that the absence of DNS resolution before a new connection is considered anomalous. This notion was first proposed by Ganger et al. (2002), and if is implemented properly, it will impose severe limitations on worm traffic. This forces scanning worms to either probe DNS namespace or issue a DNS query for each IP address, which significantly reduces the speed of worm propagation (Wong et al., 2006). Thus, it is desirable to further explore the use of DNS activities for the detection of scanning worms

Worm Detection and Containment

The proposed detection and containment mechanism, termed NEDAC (Network Detection and Data-link Containment), uses the absence of DNS query prior to contacting new destinations by a hosts. Many fast scanning worms generate pseudo-random IP addresses and attempt to make contact to find susceptible hosts. This behaviour obviates the need for DNS lookup, which is abnormal for the vast majority of legitimate publicly available services and is therefore a tell-tale sign of scanning worm propagation (Ganger et al., 2002).

The NEDAC mechanism consists of two main sub-systems that work together to provide a countermeasure solution. The first system is the network layer detection system and the second system is the data link layer containment system, with a connection maintained between the two components to enable continuous data transmission. Fig. 1 shows the flow diagram of the network layer detection system and Fig. 2 shows the flow diagram of the containment system

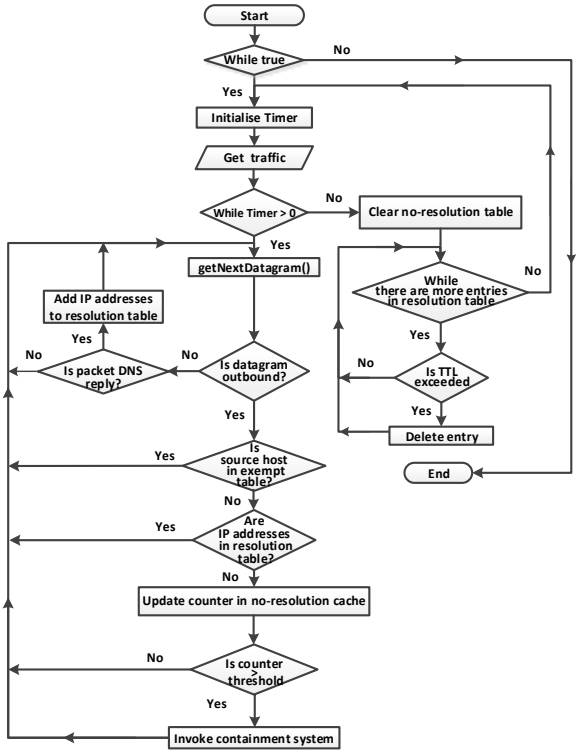

Fig. 1: Detection system

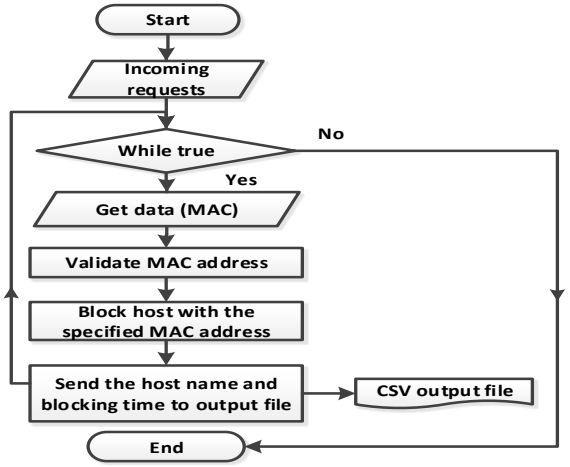

Fig. 2: Containment system

Evaluation

This section presents an evaluation of the NEDAC mechanism. Initially, a description of the methodology used to evaluate NEDAC using developed worm outbreak scenarios was presented. Then the section presents the parameters used for the worm outbreak scenarios and the experimental results obtained

To develop and use a worm in experiment, some important metric are required such the susceptible population of the worm under study, the worm datagram size and scan rates. Slammer worm is the fastest spreading worm experienced on the Internet (Moore et al., 2003). Moore et al. (2003) reported that Slammer worm had a susceptible population of 75,000 hosts and spread without payload. The authors also noted that Slammer exhibited an

average scan rate of 4000 datagrams per infected host per second and had a datagram size of 404 bytes. Thus, these metrics were used to develop a pseudo-Slammer worm outbreak scenario. Based on the susceptible population value of the Slammer worm along with the size of routable IPv4 address space (3,673,309,759 (Cotton & Vegoda, 2010)), the number of susceptible hosts per million Internet hosts for Slammer is $\left[\left(\frac{75,000}{3,673,309,759}\right) * 1,000,000\right] = 21.$

METHODOLOGY

The NEDAC mechanism has been implemented as a software prototype using the C programming language. The prototype was deployed and tested in a virtualised network testbed reported by Ahmad & Woodhead (2015). The detection system was installed on the gateway of each network and the containment system on the virtual switches in the testbed. Fig. 3 depicts the deployment of the detection and containment systems across of local network across two enterprise networks.

The pseudo-Slammer worm propagation was experimented using a worm daemon (Shahzad & Woodhead, 2014b) that has been developed with the capabilities of facilitating a worm attack event using chosen worm characteristics. The worm daemon system consists of both client and server modules capable of sending and receiving UDP datagrams. The client module is used to initiate a worm attack against the desired targets. The hosts were made susceptible to attack by running the server module, which listens on a specific UDP port and then, after receiving an "infection" datagram, continuously transmits "infectious" UDP datagrams.

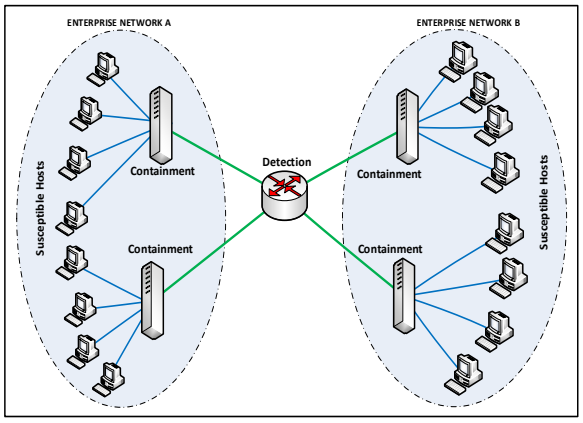

Fig. 3: Virtual environment for countermeasure testing

Upon infection, a susceptible host will send its time stamp and IP address information to the logging server for record management. The logging server has been configured with a logging daemon that keeps the details of infected host addresses and infection time. This process continues until full infection is achieved based on the details recorded on the logging server. The worm infection event was initiated by sending a UDP datagram to one of the susceptible hosts. A UDP-based worm has been chosen due to its higher rate of propagation compared to a TCP-based counterpart. UDP-based worms require no acknowledgement and cannot be detected by mechanisms that rely on number or state of failed connection attempts.

Experiments

The pseudo-Slammer experiment was conducted using 21 susceptible hosts per million Internet hosts in three class A size networks, and therefore contained $\left[2^{24} * 3 * \frac{21}{1,000,000}\right]$ = 1057 susceptible hosts. The pseudo-Slammer worm daemon was configured to listen on UDP port 1434 and then randomly transmits UDP datagrams to port 1434 at a scan rate of 125 "infectious" datagrams per second, once "infected". The scan rate was scaled down to 32% in order to avoid overloading server resources.

Five experiments were conducted using one initially infected host without any countermeasure in place. Fig. 4 shows the average result of the five experiments. The experiments were repeated with NEDAC mechanism in place using a range of threshold values of 100, 200, 400 and 500 distinct IP addresses contacted without prior DNS lookup. NEDAC was configured to invoke the containment system if a threshold is exceeded within time duration of 10 seconds. The worm infection was detected and contained by the NEDAC mechanism with no further infection across the entire range of NEDAC experiments conducted.

Further experiments were conducted with a hit-list (Staniford et al., 2002) of 10 and 20 hosts. In hit-list worm propagation, a pre-compiled list of susceptible hosts is used to initialize the infection. Then each infected host randomly transmits infection datagrams. The hit-list behaviour was tested using threshold values of 100, 200, 400 and 500, and a time duration of 10 seconds. The worm propagation was also detected and contained with no further infection for the hit-list of 10 and 20 hosts. Fig. 5 and Fig. 6 show the results of worm propagation using a hit-list of 20 hosts with and without the NEDAC mechanism

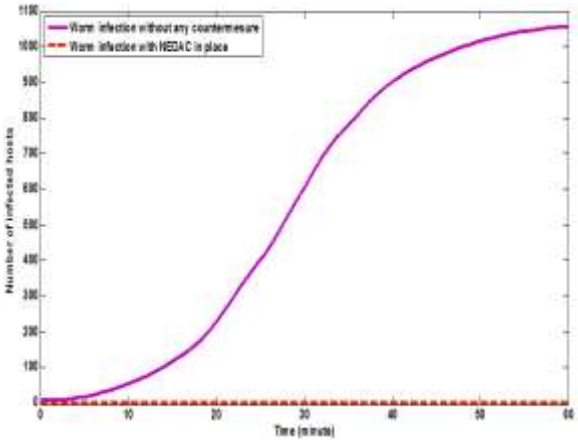

Fig. 4: Random infection behaviour

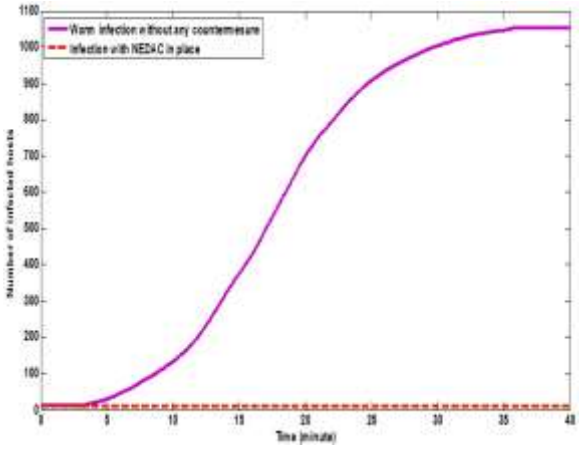

Fig. 5: Hit-list infection behaviour using 10 hosts

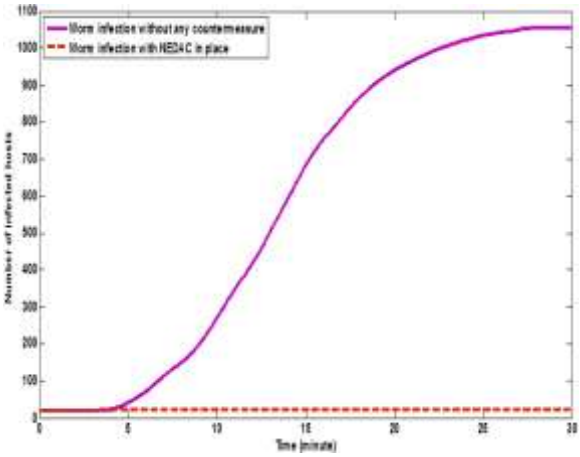

Fig. 6: Hit-list infection behaviour using 20 hosts

DISCUSSION

The experimental result for random infection without countermeasure shows that the worm attained 99% infection in 55 minutes as presented in Fig. 4. Thus using 4000 scans per second, the susceptible population of 1057 hosts could be infected in $\left[55 * \left(\frac{32}{100}\right)\right] = 18$ minutes, which is close to the infection time of Slammer (15 minutes) as reported by Moore et al. (2003). However, when the NEDAC mechanism was in place, the initially infected host was detected and contained before spreading the infection. This is achieved due to the containment solution used at the data-link layer to block all outbound datagrams from an identified infected host.

Furthermore, the result of hit-list experiment with 10 hosts shows that the worm attained 99% infection without countermeasure in 34 minutes as shown in Fig. 5. The infection time further reduced to 25 minutes with a hit-list of 20 hosts as shown in Fig. 6. In both hit-list scenarios, reduction in the times of infection were observed due to the increase in number of contacts made per second, i.e., $10 * 125 = 1250$ and $20 * 125 = 2500$ "infectious" datagrams for the hit-list of 10 and 20 hosts respectively. Despite the increased number of contacts per second, NEDAC was able to detect and contain the initially infected population of 10 and 20 hosts before spreading the infection. This was achieved due to the containment solution used at the data-link layer to isolate all outbound datagrams from an identified infected host.

Generally, the NEDAC mechanism has demonstrated effectiveness in detecting and containing fast scanning Internet worms at an early stage.

Conclusion and Future Work

This article has presented a mechanism that uses DNS activities to detect anomalies at network layer and employs a data-link layer containment system to isolate an infected host. The empirical results of the experiments conducted showed that the mechanism can detect and completely contain fast scanning Internet worms including hit-list worm propagation scenario. This is due to the containment techniques employed in the data link layer that isolates a given infected host from the network and therefore ends the worm propagation.

As future work, it is desirable to further optimise the mechanism, particularly the detection scheme. The mechanism will further be evaluated using background traffic to test the effects of false alarms. The complexity of the detection system will be evaluated and then a comparative evaluation of the mechanism with existing worm detection techniques will be conducted.

REFERENCES

Ahmad, Muhammad Aminu, & Woodhead, Steve. 2015 (September). Containment of Fast Scanning Computer Network Worm. Pages 235 – 247 of: 8th International Conference, IDCS 2015. Springer.

Cotton, M, & Vegoda, L. 2010. Special Use IPv4 Addresses. Tech. rept. BCP 153, RFC 5735, January. CVE. 2014. Common Vulnerabilities and Exposures. [Online]. Accessed on 19th October 2014. Avalable: https://cve.mitre.org/.

Fosnock, Craig. 2005. Computer worms: past, present, and future. East Carolina University, 8.

Ganger, Gregory R, Economou, Gregg, & Bielski, Stanley M. 2002. Self-Securing Network Interfaces: What, Why and How? Tech. rept. DTIC Document.

Garcia-Teodoro, Pedro, Diaz-Verdejo, J, Maciá-Fernández, Gabriel, & Vázquez, Enrique. 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. computers & security, 28(1), 18–28.

Gu, Guofei, Sharif, Monirul, Qin, Xinzhou, Dagon, David, Lee, Wenke, & Riley, George. 2004. Worm detection, early warning and response based on local victim information. Pages 136–145 of: Computer Security Applications Conference, 2004. 20th Annual. IEEE.

Joukov, Nikolai, & Chiueh, Tzi-cker. 2003. Internet worms as internet-wide threat. Experimental Computer Systems Lab, Tech. Rep. TR-143, September.

Jung, Jaeyeon, Paxson, Vern, Berger, Arthur W, & Balakrishnan, Hari. 2004. Fast portscan detection using sequential hypothesis testing. Pages 211–225 of: Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on. IEEE.

Jyothsna, V, Prasad, VV Rama, & Prasad, K Munivara. 2011. A review of anomaly based intrusion detection systems. International Journal of Computer Applications, 28(7), 26–35.

Li, Pele, Salour, Mehdi, & Su, Xiao. 2008. A survey of internet worm detection and containment. Communications Surveys & Tutorials, IEEE, 10(1), 20–35.

Moore, David, Paxson, Vern, Savage, Stefan, Shannon, Colleen, Staniford, Stuart, & Weaver, Nicholas. 2003. Inside the slammer worm. IEEE Security & Privacy, 33–39.

Niemelä, Jarno, & Palomäki, Pirkka. 2013 (Nov. 19). Malware detection by application monitoring. US Patent 8,590,045.

NVD. 2014. National Vulnerability Database. http://nvd.nist.gov/.

Rasheed, Mohammad M, Norwawi, Norita Md, Ghazali, Osman, & Kadhum, Mohammed M. 2009. Intelligent failure connection algorithm for detecting internet worms. IJCSNS, 9(5), 280.

Shahzad, Khurram, & Woodhead, Steve. 2014a. Towards automated distributed containment of zero-day network worms. Pages 1–7 of: Computing, Communication and Networking Technologies (ICCCNT), 2014 International Conference on IEEE.

Shahzad, Khurram, & Woodhead, Steve. 2014b. A Pseudo-Worm Daemon (PWD) for empirical analysis of zero-day network worms and countermeasure testing. Pages 1–6 of: Computing, Communication and Networking Technologies (ICCCNT), 2014 International Conference on. IEEE.

Smith, Craig, Matrawy, Ashraf, Chow, Stanley, & Abdelaziz, Bassem. 2009. Computer worms: Architectures, evasion strategies, and detection mechanisms. Journal of Information Assurance and Security, 4, 69–83.

Staniford, Stuart, Paxson, Vern, Weaver, Nicholas, et al. 2002. How to Own the Internet in Your Spare Time. Pages 149–167 of: USENIX Security Symposium.

Tidy, Luc, Shahzad, Khurram, Aminu, Ahmad Muhammad, & Steve, Woodhead. 2014 (October). An assessment of the contemporary threat posed by network worm malware. In: The Ninth Internation Conference on Systems and Networks Communications (ICSNC 2014).

Weaver, Nicholas, Staniford, Stuart, & Paxson, Vern. 2004. Very Fast Containment of Scanning Worms. Pages 16–85 of: USENIX Security Symposium, vol. 2.

Whyte, David, Kranakis, Evangelos, & van Oorschot, Paul C. 2005. DNS-based Detection of Scanning Worms in an Enterprise Network. In: NDSS.

Williamson, Matthew M. 2002. Throttling viruses: Restricting propagation to defeat malicious mobile code. Pages 61–68 of: Computer Security Applications Conference, 2002. Proceedings. 18th Annual. IEEE.

Wong, Cynthia, Bielski, Stan, Studer, Ahren, & Wang, Chenxi. 2006. Empirical analysis of rate limiting mechanisms. Pages 22–42 of: Recent Advances in Intrusion Detection. Springer.